ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/384249534

Case studies of common csharp project report

Research Proposal · September 2024

DOI: 10.13140/RG.2.2.20281.99681

CITATIONS
0

READS
373

1 author:

Kamal Acharya
Tribhuvan University
249 PUBLICATIONS **4,439** CITATIONS

SEE PROFILE

All content following this page was uploaded by Kamal Acharya on 23 September 2024.

The user has requested enhancement of the downloaded file.

Chapter 1 Case Study

Problem Statement

WebDev Enterprises is a medium sized company that develops Web applications using J2EE technology for its overseas clients such as BlueSoft Software Solutions and InterSolutions Private Limited. The management at WebDev recently conducted an analysis and discovered that it has not been able to use the code of its existing applications developed in J2EE when developing new J2EE applications. As a result, the deadlines for the new projects were not being met causing financial loss to WebDev Enterprises. Because of not being able to meet deadlines, WebDev Enterprises was losing its clients. In addition, the management of WebDev Enterprises found that it was a complex task to develop applications in J2EE. How can the management of WebDev solve the problem?

Solution

In order to be able to reuse the code of existing applications, WebDev must start creating Web based applications using C# and ASP.NET. C# provides features such as inheritance and interface that allow reusability of code. This means that code written for a specific application in C# can be easily used for another C# application. Inheritance feature of C# allows a class to inherit properties, which contain data and methods of another class. For example, if you create a class A and define certain properties such as name and address in it, and a display method to display the values contained in the properties, then you can use the properties and methods of A class in another class B.

Interfaces in C# allow you to declare abstract methods. These interfaces can be implemented in multiple classes. Each class in which the interface is implemented can contain specific code for the abstract methods defined in the interface. Thus, with the use of C# features such as Inheritance and Interface, code of one C# application can be reused in another C# application. As a result, development time for creating an application in C# is reduced.

With the use of C#, programmers working at WebDev Enterprises could develop applications efficiently and in less time.

C# programming language is supported by Microsoft Visual Studio 2005, which is a rapid development tool for developing Windows based and Web based applications. The support for .NET Framework in C# allows you to create interactive user interfaces for applications in less time. The .NET Framework provides the Control class that helps use controls, their properties and methods to develop interactive user interfaces for an application in less time.

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

Because of all these reasons, which include C# support in Microsoft Visual Studio 2005 and inheritance, the programmers working in WebDev were able to meet the deadlines for the new projects and satisfy the requirements of the overseas clients.

ASP.NET is a technology supported by Microsoft Visual Studio 2005. It helps in rapid development of Web based applications. ASP.NET also supports C# programming language for development of Web based applications. In ASP.NET, programmers at WebDev Enterprises can use the controls available in Microsoft Visual Studio 2005 through the Microsoft .NET framework. The controls available in Visual Studio 2005 and supported by ASP.NET can be used to create graphical user interface for Web based applications in less amount of time.

Thus, with the use of C# and ASP.NET, WebDev has been able to solve its problem and rapidly develop applications for its overseas clients.

Chapter 3 Case Study

Problem Statement

AUPart Enterprise is a manufacturing unit that is involved in the manufacturing of automobile parts. This company was established in 1980 and since then the staff of AuPart has been manually keeping the records of manufacturing parts. Now, it has been discovered by the management of AUPart Enterprise that manual record keeping results in occurrence of a large number of errors. In addition, it has also been discovered that manually keeping records of manufactured parts is time consuming. How can management of AuPart Enterprise automate the task of maintaining records of manufactured parts?

Solution

After much research, the management of AUPart Enterprise comes to the conclusion that an application created in a high level programming language such as C# is needed to automate the task of maintaining the records of manufactured parts. Quick Soft Solutions Private Limited, which is a software development company, is contacted by AUPart Enterprise to develop the application for automating the task of maintaining the records of manufactured parts. The team of programmers at Quick Soft Solutions Private Limited first conducted an interview with the staff of AUPart Enterprise to find out their requirements. After requirement analysis, it was decided by the programmers at Quick Soft Solutions Private Limited that C# features such as constructors and variables must be used in the application for automating the task of keeping records of manufactured parts:

```
using System;
using System.Collections;

class Inventory {
  string partName;
  double partPrice;
  int priceQty;

  public Inventory(string pName, double pPrice, int pQty) {
    partName = pName;
    partPrice = pPrice;
    priceQty = pQty;

    Console.WriteLine("\nPART NAME  : {0,-10}",partName);
    Console.WriteLine("\tPrice   : Rs.{0,5} \n\tQuantity:
{1}",partPrice, priceQty);
  }
}

public class StockData {
  public static void Main() {
    Console.WriteLine("==ABC Spare Parts Ltd. Stock List==");
    new Inventory("Bolts", 2.25, 100);
    new Inventory("Shockers", 5.25, 30);
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
        new Inventory("Brake Pads", 3.50, 50);
        new Inventory("Steering", 120, 20);

    }
}
```

In C#, constructors help initialise the objects of a class and variables are used to store data related to an object. The name of the constructor in C# must be same as the name of the class.

Chapter 5 Case Study

Problem Statement

Martin is working as a junior programmer in HighFi Software Solutions, which is a software development company involved in developing applications in C, Java and C# programming languages. The Project Manager of Martin has asked him to create an application for converting a number into its equivalent binary format. Martin has to develop the application for an overseas client, TechSoft Solutions. How should Martin create the application?

Solution

To satisfy the overseas client, Martin should use an object oriented language such as C# to create the application for converting a number into binary format. C# provides bitwise operators that can be easily used in a C# application to perform manipulation on binary data. After much research, Martin develops the following application in C# for automating the task of converting a number into binary format:

```
using System;

class Convert {
  public int nobits;

  public Convert(int n) {
    nobits = n;
  }

  public void ConvertToBinary(int num) {
    int mask = 1;
    mask <<= nobits-1;

    int spaceVal = 0;
    for(; mask != 0; mask >>= 1)
    {
      if((num & mask) != 0)
           Console.Write("1");
      else
           Console.Write("0");
      spaceVal++;
      if((spaceVal % 8) == 0)
      {
        Console.Write(" ");
        spaceVal = 0;
      }
    }
    Console.WriteLine();
  }
}

public class BitWiseDemo {
  public static void Main() {
```

```
        Console.WriteLine("Enter a Number for Conversion to Binary Form: ");
        int num=int.Parse(Console.ReadLine());

        Convert con = new Convert(8);
        Console.WriteLine("The Number in Binary Form is: ");
        con.ConvertToBinary(num);

    }
}
```

The bitwise operators can be easily used in a C# application to shift the bits from left to right and vice versa. Manually performing the task of bit manipulation is time consuming and error prone.

Chapter 7 Case Study

Problem Statement

St Peters School is a convent school that provides education from K.G class to 12th class. Quarterly, half yearly and final exams are conducted every year for students of all classes in St Peters School. The class teacher of each class has to prepare a student report for every student after a particular exam has been conducted. Recently, it has been found by the principal of St Peters School that a lot of time is being taken by the class teacher in preparing the student reports. The principal has also found out that the efficiency of the teachers have decreased because they have been manually performing the task of preparing the student reports for each student. St Peters have 600 students studying in its different classes. Preparing student report for at least 30 students per class has become a tedious task for the class teacher of a specific class. How can the principal of St Peters School solve the problem of the teachers?

Solution

After holding meetings with senior teachers of the school, the principal of St Peters School has come to the conclusion that the task of calculating total marks for preparing a student report must be automated through a software application running on a computer system. After coming to this conclusion, the principal of St Peters School asks Nandita, who is teaching computer science to students of higher classes such as 9th , 10th, 11th and 12th to create the application for automating the task of calculating total marks that is performed by class teachers while preparing student report. For this, Nandita first conducts an interview with the class teachers of different classes to understand their requirements. After the analysis of the requirements, Nandita decides to use decision making and looping concept of C# in an application to automate the task of calculating the total marks for each student of a class. Decision making and looping allows a programmer to decide the flow of control for a program or application. Nandita creates the following application for automating the task of calculating total marks:

```
using System;

class StudentMarks
{
      static void Main(string[] args)
      {
       int[] marks = new int[5];
       int totalMarks = 0;

       Console.WriteLine("==Student Information==\n");

       for (int i = 0; i < 5; i++)
       {
        Console.WriteLine("Enter Your Marks In Subject {0}: ",i+1);
        marks[i] = int.Parse(Console.ReadLine());
       }
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
        Console.WriteLine();

        // Displaying Student Marks
        Console.Write("The marks you have entered are: ");
        for (int i = 0; i < 5; i++)
        {
         if(i<4)
            Console.Write(marks[i] + ", ");
         else
            Console.Write(marks[i]);
        }

        // Calculating and Displaying Total Marks of the Student
        for (int i = 0; i < 5; i++)
        {
         totalMarks += marks[i];
        }

        Console.WriteLine();
        Console.WriteLine("Your total marks are: {0}", totalMarks);

        // Calculating and Displaying Maximum Marks of the Student
        int maxMarks = marks[0];
        int index=1;

        for (int i = 1; i < marks.Length; i++)
        {
         if (marks[i] > maxMarks)
         {
          maxMarks = marks[i];
          index=i+1;
         }
        }

        Console.WriteLine("You have scored the maximum marks {0} in
Subject {1}.", maxMarks,index);

        // Calculating and Displaying Minimum Marks of the Student
        int minMarks = marks[0];
        index=1;

        for (int i = 1; i < marks.Length; i++)
        {
         if (marks[i] < minMarks)
         {
          minMarks = marks[i];
          index=i+1;
         }
        }

        Console.WriteLine("You have scored the minimum marks {0} in
Subject {1}.", minMarks,index);

        // Calculating and Displaying Percentage of the Student
        double percentage;
        double val=0.0;
```

```
        val=(double)totalMarks / 500;
        percentage = (double) val*100;
        Console.WriteLine("Your percentage is: {0}%", percentage);
        Console.ReadLine();

       }

}
```

In looping, a set of program staements are executed based on a condition. The program statements are executed until a specific condition is true. When the condition becomes false the exeuction of the program statements is terminated.

Chapter 9 Case Study

Problem Statement

B.G International School is an educational institute that provides primary education to children from K.G class to 10th class. The administrative staff of B.G International School has to maintain the records, which include name, class and rollno of students studying in different classes. Manually maintaining the record of each student is a tedious process for the administrative staff of B.G International School. A large number of errors occur when the administrative staff enters the records manually in registers. How can B.G International School automate the task of entering student records?

Solution

In order to make the process of maintaining student records easy and efficient, B.G International School contacts SoftDev software development organisation to create an application, which will help in the task of entering the student records. SoftDev assigns the task of creating the application for B.G International School to Siddarth, who is a senior programmer in the organisation. Siddarth decides to use the C# programming language for creating the application to enter student records. C# provides the ArrayList class in the System.Collections namespace. This ArrayList class allows you to create an array list, which can store array objects. The size of the array list is fixed at run time. Thus the array list created using ArrayList class is a dynamic array list. Because a large amount of data has to be entered in case of B.G International School, use of array list will be helpful for Siddarth.

After understanding the requirements of B.G International School, Siddarth develops the following C# application

```
using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;

namespace AddStudentDetails
{
    class StudentDetails
    {


        static void Main(string[] args)
        {
            string studname1, studname2, studname3;
            string classname1, classname2, classname3;
            string rollno1, rollno2, rollno3;
            string getresult;
            ArrayList arrlist = new ArrayList();
            Console.WriteLine("The initial capacity of arraylist =" +
arrlist.Capacity);
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
            Console.WriteLine("The initial elements are =" +
arrlist.Count);

            Console.WriteLine("Enter Details of Student 1");
            Console.WriteLine("_____________________________");
            Console.Write("Enter Name of Student1 : ");
            studname1 = Console.ReadLine();
            Console.Write("Enter Class of Student1 : ");
            classname1 = Console.ReadLine();
            Console.Write("Enter Roll No. of Student1 : ");
            rollno1 = Console.ReadLine();
            Console.WriteLine("_____________________________");
            Console.Write("Enter Name of Student2 : ");
            studname2 = Console.ReadLine();
            Console.Write("Enter Class of Student2 : ");
            classname2 = Console.ReadLine();
            Console.Write("Enter Roll No. of Student2 : ");
            rollno2 = Console.ReadLine();
            Console.WriteLine("_____________________________");
            Console.Write("Enter Name of Student3 : ");
            studname3 = Console.ReadLine();
            Console.Write("Enter Class of Student3 : ");
            classname3 = Console.ReadLine();
            Console.Write("Enter Roll No. of Student3 : ");
            rollno3 = Console.ReadLine();
            Console.WriteLine("_____________________________");

            arrlist.Add(studname1);
            arrlist.Add(classname1);
            arrlist.Add(rollno1);
            arrlist.Add(studname2);
            arrlist.Add(classname2);
            arrlist.Add(rollno2);
            arrlist.Add(studname3);
            arrlist.Add(classname3);
            arrlist.Add(rollno3);

            Console.WriteLine("Current Capacity=" + arrlist.Capacity);
            Console.WriteLine("ArrayList having elements = " +
arrlist.Count);
            Console.Write("ArrayList is having content:-");
            for (int i = 0; i < arrlist.Count; i++)
            {
                Console.Write(arrlist[i] + " ");

            }

            Console.WriteLine("\n");
            Console.WriteLine("To remove 2nd student details press y or
to interchange student details1 with student details3 press n.");
            getresult = Console.ReadLine();
            if (getresult == "y")
            {
                Console.WriteLine("Removing 2nd Student details");
                Console.WriteLine("_____________________________");
                arrlist.Remove(studname2);
                arrlist.Remove(classname2);
```

```
                arrlist.Remove(rollno2);
                Console.WriteLine("Current Capcity of the arraylist is
now =" + arrlist.Capacity);
                Console.WriteLine("Arraylist is having elements=" +
arrlist.Count);
                Console.Write("The elements of Arraylist are now:-");
                for (int m = 0; m < arrlist.Count; m++)
                {
                    Console.Write(arrlist[m] + " ");
                }
                Console.WriteLine("\n");
            }
            else
            {
               Console.WriteLine("\n");
               Console.WriteLine("Changing elements");
               Console.WriteLine("_____________________________");
               arrlist[0] = studname3.ToString();
               arrlist[1] = classname3;
               arrlist[2] = rollno3;
               arrlist[6] = studname1;
               arrlist[7] = classname1;
               arrlist[8] = rollno1;

               Console.Write("Now the elements of the ArrayList are :
");
               for (int m = 0; m < arrlist.Count; m++)
               {
                   Console.Write(arrlist[m] + " ");
               }
               Console.WriteLine("\n");

            }
        }

    }
}
```

With the use of C# application created by Siddarth, the administrative staff of B.G International will be able to save time in entering the details such as name and class of students studying in different classes.

Chapter 11 Case Study

Problem Statement

CompTech is a medium sized organisation that is involved in sale of computer hardware. It also has a small software development department, which develops customised software and Web sites for different organisations. CompTech has been till now manually maintaining the records of its employees, which is time consuming and decreases the efficiency of the H.R department staff. . How can CompTech solve the problem?

Solution

CompTech requests one of the programmer, Harsh, who has been working for past two years in the software development department, to create an application, which will enable the H.R department to automate the task of entering employee data. In order to judiciously utilise memory space, Harsh decides to use structure feature of C#. A structure is a value type that allows you to organise related data of different data types. After much research, Harsh creates the following C# application:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace StructureExample
{
    public struct Employee
    {
        private string emp_details;

        public string Write
        {
            get { return emp_details; }
            set { emp_details = value; }
        }

        public string Read()
        {
            return Console.ReadLine();
        }
    }

    public struct EmployeeDetails
    {
        Employee name;
        Employee address;
        Employee telephone;
        Employee dept;
        Employee sal;

        public Employee Name
        {
            get { return name; }
```

```
            set { name = value; }
        }

        public Employee Address
        {
            get { return address; }
            set { address = value; }
        }

        public Employee Telephone
        {
            get { return telephone; }
            set { telephone = value; }
        }

        public Employee Department
        {
            get { return dept; }
            set { dept = value; }
        }

        public Employee Salary
        {
            get { return sal; }
            set { sal = value; }
        }

        public void ShowAllDetails()
        {
            Employee emp = new Employee();
            Console.WriteLine("Employee Details Registration System.");
            Console.WriteLine("-------------------------------------------------");
            Console.Write("Enter the Name of the Employee: ");
            name.Write = emp.Read();
            Console.Write("Enter the Address of the Employee: ");
            address.Write = emp.Read();
            Console.Write("Enter the Telephone of the Employee: ");
            telephone.Write = emp.Read();
            Console.Write("Enter the Department of the Employee: ");
            dept.Write = emp.Read();
            Console.Write("Enter the Salary of the Employee: ");
            sal.Write = emp.Read();

        }
    }

    class ShowEmployeeDetails
    {
        static void Main(string[] args)
        {
            string input = "";
            EmployeeDetails emp1 = new EmployeeDetails();

            emp1.ShowAllDetails();
            Console.WriteLine();

            Console.WriteLine("Showing Employee Details");
```

```
            Console.WriteLine(" ----------------------------------------------");
            Console.WriteLine("Name:    {0}", emp1.Name.Write);
            Console.WriteLine("Address:    {0}", emp1.Address.Write);
            Console.WriteLine("Telephone: {0}", emp1.Telephone.Write);
            Console.WriteLine("Department: {0}", emp1.Department.Write);
            Console.WriteLine("Salary:      {0}", emp1.Salary.Write);
            Console.WriteLine(" ----------------------------------------------");
            Console.WriteLine("Press y to save the record. ..... ");
            input = Console.ReadLine();
            if (input == "y")
            {
                Console.WriteLine("Record Saved.");
            }
            else
            {
                Console.WriteLine("Record not saved.");
            }

        }
    }
}
```

The use of structure in a C# application allows you to manage the memory of a computer efficiently as all related data is organised in a structure. It also increases the performance of a C# application.

Chapter 13 Case Study

Problem Statement

ArchBuild is a small organisation, which is involved in creating architectural designs for buildings such as homes, hospitals and educational institutes. While making the architectural designs, the architects working in ArchBuild need to calculate the area of different shapes such as rectangle, circle and square. Till now, these architects have been calculating areas manually using calculator for complex calculations. Manual calculation of different shapes is time consuming. As a result, the efficiency of the architects working in ArchBuild has decreased considerably. How can ArchBuild solve this problem?

Solution

To automate the task of calculating area of different shapes such as circle and rectangle, ArchBuild contacts ABC SoftDev Private Limited, which is a software development organisation. ArchBuild asks ABC SoftDev to create an application, which will allow the architects working in ArchBuild to obtain the area of a shape. ABC SoftDev first performs requirement analysis to understand the requirements of the architects. After requirement analysis, it starts developing an application using the inheritance feature of C# programming language. Inheritance feature allows a class in C# to inherit the properties and methods defined in another class besides having its own properties and methods. Inheritance allows programmers to reuse the code of one application in another application. The ABC SoftDev creates the following application for ArchBuild to automate the task of calculating area for different shapes:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace CaseStudy13
{
    public abstract class Dimension_Shape
    {

        protected string colorValue;

        public Dimension_Shape(string color)
        {
            this.colorValue = color;
        }

        public string getColor()
        {
            return colorValue;
        }

        public abstract double getAreaof_Shape();


    }
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
    public class ShapeCircle : Dimension_Shape
    {

        private double radiusofCircle;

        public ShapeCircle(string colorValue, double radius)
            : base(colorValue)
        {
            this.radiusofCircle = radius;
        }

        public override double getAreaof_Shape()
        {
            return System.Math.PI * radiusofCircle * radiusofCircle;
        }
    }

    public class ShapeRectangle : Dimension_Shape
    {

        private double lengthofRectangle;
        private double widthofRectangle;

        public ShapeRectangle(string colorValue, double length, double
width)
            : base(colorValue)
        {
            this.lengthofRectangle = length;
            this.widthofRectangle = width;
        }

        public override double getAreaof_Shape()
        {
            return lengthofRectangle * widthofRectangle;
        }
    }

    public class ShapeSquare : Dimension_Shape
    {

        private double lengthofSquare;

        public ShapeSquare(string colorValue, double length) :
base(colorValue)
        {
            this.lengthofSquare = length;
        }

        public override double getAreaof_Shape()
        {
            return lengthofSquare * lengthofSquare;
        }
    }


    class AreaandColor
```

```
    {
        static void Main(string[] args)
        {
            Console.WriteLine("This is an example where all the
properties of different classes\nlike Circle,Square can be defined in
another class and can be inherited \nfrom it when required without
populating the classes with excess codes.");
            Console.WriteLine("----------------------------");
            Dimension_Shape myCircle = new ShapeCircle("Maroon", 6);
            Dimension_Shape myRectangle = new ShapeRectangle("Blue",
12, 9);
            Dimension_Shape mySquare = new ShapeSquare("Gray", 15);
            System.Console.WriteLine("The Circle is of " +
myCircle.getColor()
                + " color and its area is " +
myCircle.getAreaof_Shape() + ".");
            Console.WriteLine("----------------------------");
            System.Console.WriteLine("The Rectangle is of " +
myRectangle.getColor()
              + " color and its area is " +
myRectangle.getAreaof_Shape() + ".");
            Console.WriteLine("----------------------------");
            System.Console.WriteLine("The square is of " +
mySquare.getColor()
              + " color and its area is " + mySquare.getAreaof_Shape()
+ ".");
            Console.WriteLine("----------------------------");

        }
    }
}
```

The use of inheritance in the application for calculating the area helps programmers avoid defining excess code in one class. The properties and methods defined in one class can be easily used in another class because of inheritance

Chapter 15 Case Study

Problem Statement

UInfo Private Limited is involved in the development of new aircrafts for companies like AeroFly and FlyHigh. When developing the designs for the new aircrafts the designing staff of UInfo has to do a number of complex calculations. Manually performing these complex calculations has become tedious task for the designing staff of UInfo. How can UInfo automate the task of performing complex calculations?

Solution

The management of UInfo after much discussion came to the conclusion that an application must be developed to automate the task of performing complex calculations. For this purpose, it contacts a software development company called ABC Software Solutions Private Limited. This software development company first performs a requirement analysis to understand the requirements of UInfo. It then decides that operator overloading feature of C# should be used in a C# application to overload operators such as + and _ so that different functions can be performed using the same operator. After 10 days, the ABC Software Solutions Private Limited comes up with the following C# application, which helps perform mathematical operations such as adding and multiplying two numbers:

```
//Operator Overloading using inheritance

using System;
using System.Collections.Generic;
using System.Text;

namespace OperatorOverloading
{

    class StoreValue
    {
        private int num1;
        private int num2;
        public StoreValue()
        {
        }
        public StoreValue(int m, int n)
        {
            num1 = m+m;
            num2 = n+n;
        }
        public void DisplayWindow()
        {
            Console.WriteLine("The sum of numbers is {0} {1}", num1,
num2);
        }
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
    }

    class InheritValue : StoreValue
    {
        private double num1 ;
        private double num2 ;
        public InheritValue(double a, double b)
        {
            num1 = a*a;
            num2 = b*b;
        }
        public InheritValue()
        {
        }
        public new void DisplayWindow()
        {
        Console.WriteLine("Now the square of the numbers 355 and 455
are {0} and {1}",num1,num2);
        }
    }

    class InheritMoreValue : InheritValue
    {
        private double num1;
        private double num2;
        public InheritMoreValue(double a, double b)
        {
            num1 = (a * b) / (a+b);
            num2 = (a * b) / (b-a);
        }
        public InheritMoreValue()
        {
        }
        public new void DisplayWindow()
        {
            Console.WriteLine("The value of the numbers 48 and 62 are
{0} and {1}", num1, num2);
        }
    }

    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Operator Overloading with Inheritance.");
            StoreValue Obj1 = new StoreValue(119,280);
            Obj1.DisplayWindow();
            InheritValue Obj2 = new InheritValue(355,455);
            Obj2.DisplayWindow();
            InheritMoreValue Obj3 = new InheritMoreValue(150,250);
            Obj3.DisplayWindow();

        }
    }
}
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

By the use of operator overloading in a C# application, a new definition for operators such as +, _ and ++ can be provided. Operator overloading allows you to use operators supported by C# to work with classes and structs in the same way as these operators function with built-in data types.

Chapter 17 Case Study

Problem Statement

Modern Engineering Collage provides courses for streams such as mechanical, electrical, electronics and civil. When a student takes admission in the Modern Engineering Collage and comes to the administrative staff to pay the fees, the staff has to manually fill a form specifying details such as name, address, course and fees. A large amount of time is consumed in the filling of the form. In addition, the efficiency of the administrative staff also decreases as a lot of time is spent in filling of the form. The students who are taking admission in the Modern Engineering Collage have to wait for long time in a queue for paying their fees. How can Modern Engineering Collage solve this problem of the students and the administrative staff?

Solution

The principal of Modern Engineering Collage along with some of the important management staff conducts a meeting and tries to find the solution to the problem, which is being faced by the students taking admission in the collage and the administrative staff. By the end of the meeting, the principal and the management staff come to the conclusion that an application must be developed for automating the task of entering student information such as name and address. The principal calls the Head of the Electronics department after a few days and asks the Head to develop the application. The Head of the Electronics department first conducts a requirement analysis through interviews with the administrative staff of the collage. After the analysis, the Head of Electronics Department comes to the conclusion that methods such as ReadLine() and WriteLine() available in C# must be used in a C# application for obtaining details such as name and address and displaying a message to confirm the admission of the student. The methods such as ReadLine() and WriteLine() help in receiving inputs from users and displaying output. The following C# application is developed by the Head of the Electronics department:

```
using System;
using System.Collections.Generic;
using System.Text;

namespace StudentAdmissionSystem
{
     public enum Degree
    {
        Mechanical = 1,
        Electrical,
        Electronics,
        Aeronautics,
        Civil
    }

    class Select
    {
```

```
        public Degree degree;

        public Select()
        {
            degree = Degree.Civil;
        }
        public Select(Degree degree)
        {

            degree = Degree.Civil;
        }

    }

    class GetDetails
    {
        Select degree;
        string name,address,telephone,course,fees;
        public GetDetails()
        {
            degree = new Select();
        }


        public void GetAdmission()
        {
            int choice = 0;

            Console.WriteLine("Enter the Degree course to take
admission in Modern Engineering College : ");
            Console.WriteLine(" ----------------------------------------");
            Console.WriteLine("1. Mechanical");
            Console.WriteLine("2. Electrical");
            Console.WriteLine("3. Electronics");
            Console.WriteLine("4. Aeronautics");
            Console.WriteLine("5. Civil");

            Console.Write("Please enter your Choice: ");
            choice = int.Parse(Console.ReadLine());

            switch (choice)
            {
                case 1:
                    degree.degree = Degree.Mechanical;
                    break;
                case 2:
                    degree.degree = Degree.Electrical;
                    break;
                case 3:
                    degree.degree = Degree.Electronics;
                    break;
                case 4:
                    degree.degree = Degree.Aeronautics;
                    break;
                default:
                    degree.degree = Degree.Civil;
                    break;
```

```
            }
        }

        public void GetStudentDetails()
        {
            Console.Write("Enter name : ");
            name = Console.ReadLine();
            Console.WriteLine("");
            Console.Write("Enter address : ");
            address = Console.ReadLine();
            Console.WriteLine("");
            Console.Write("Enter telephone : ");
            telephone = Console.ReadLine();
            Console.WriteLine("");
            Console.WriteLine("");
            Console.Write("Course : " + degree.degree);
            course = degree.degree.ToString();
            Console.WriteLine("");
            Console.Write("Enter Fees : ");
            fees = Console.ReadLine();
        }




        class StudentMenu
        {
            static void Main(string[] args)
            {
                GetDetails gd = new GetDetails();
                gd.GetAdmission();

                Console.WriteLine();

                gd.GetStudentDetails();
                Console.WriteLine();
                Console.WriteLine("Thank You " + gd.name + " for
getting Admission in " + gd.course+ " course.");
            }
        }
    }
}
```

In C#, methods such as Read() and Write() allow programmers to perform input and output operations. This means that these methods can be used to receive inputs such as text and numbers from a user in a specific format and display the output in a particular format.

Chapter 19 Case Study

Problem Statement

Nishant is working as a programmer for Best IT Solutions Private company, which is involved in developing high end software applications using technologies and programming languages, such as ASP.NET and C#. Nishant is asked by his Project Manager to create a server based application for B.M Enterprises. The server based application has to run on a server for processing of specific data. The Project Manager has asked Nishant to take care that the load on the server must be minimum when the application is run. What functionality should Nishant implement in his application to decrease the load on server?

Solution

Nishant must implement the functionality of multithreading, which is supported in C# programming language to decrease the load on the server. In multithreading, multiple threads are run simultaneously during the execution of a program or application. Nishant first creates the following sample C# application to understand the concept of multithreading:

```
using System;
using System.Collections.Generic;
using System.Text;
using System.Threading;

namespace Threading
{
    class ThreadedApplications
    {
        static void Main(string[] args)
        {
            ApplicationCounter acObj = new ApplicationCounter();
            ThreadStart execute = new
ThreadStart(acObj.application_stopstart);
            ThirdPartyApplications acObj2 = new
ThirdPartyApplications();
            ThreadStart execute2 = new
ThreadStart(acObj2.thirdapplication_stopstart);
            Thread applications = new Thread(execute);
            Console.WriteLine("Starting the Primary Application.");
            Thread applications2 = new Thread(execute2);
            //starting the primary application
            applications.Start();
            applications2.Start();

            //starting the secondary applications

            for (int m = 1; m <= 5; m++)
            {
                Console.WriteLine("Executing the Primary Application:
p{0}", m);
```

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

```
                Thread.Sleep(1000);

            }
            Console.WriteLine("Primary Application ended");
        }
    }

    public class ApplicationCounter
    {

        public void application_stopstart()
        {
            Console.WriteLine("Starting Secondary Applications now.");
            for (int m = 1; m <= 10; m++)
            {
                Console.WriteLine("Executing Secondary Application :
s{0}", m);

                Thread.Sleep(500);

            }
            Console.WriteLine("Secondary Application ended");
        }
    }

    public class ThirdPartyApplications
    {

        public void thirdapplication_stopstart()
        {
            Console.WriteLine("Starting Third Party Applications now.");
            for (int m = 1; m <= 15; m++)
            {
                Console.WriteLine("Executing Third Party Application :
t{0}", m);

                Thread.Sleep(2000);

            }
            Console.WriteLine("Third Party Applications ended");
        }
    }

}
```

In C#, multithreading helps in optimising the resources related to a specific application. It also helps in faster execution of an application. Multithreading can be used in C# to create different threads for performing operations such as checking the user input and performing background tasks.

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

References

1. *Kamal Acharya. School management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.34023165/v1*
2. *Kamal Acharya. A CASE STUDY OF CINEMA MANAGEMENT SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254873.30191075/v1*
3. *Kamal Acharya. A CASE STUDY ON ONLINE TICKET BOOKING SYSTEM PROJECT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254872.26972790/v1*
4. *Kamal Acharya. Web chatting application project report management system. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.18588592/v1*
5. *Kamal Acharya. RETAIL STORE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172254871.14590154/v1*
6. *Kamal Acharya. SUPERMARKET MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.19145062/v1*
7. *Kamal Acharya. SOCIAL MEDIA MANAGEMENT SYSTEM PROJECT REPORT. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252491.11210579/v1*
8. *Kamal Acharya. Online music portal management system project report. Authorea. August 01, 2024. DOI: https://doi.org/10.22541/au.172252488.89734698/v1*
9. *Kamal Acharya. COLLEGE BUS MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245277.70798942/v1*
10. *Kamal Acharya. AUTOMOBILE MANAGEMENT SYSTEM PROJECT REPORT. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172245276.67982593/v1*
11. *Kamal Acharya. Ludo management system project report. Authorea. July 31, 2024.DOI: https://doi.org/10.22541/au.172243999.98091616/v1*
12. *Kamal Acharya. Literature online quiz system project report. Authorea. July 31, 2024. DOI: https://doi.org/10.22541/au.172243825.53562953/v1*
13. *Kamal Acharya. Avoid waste management system project. Authorea. July 29, 2024. DOI: https://doi.org/10.22541/au.172228528.85022205/v1*

Visit researchgate profile : https://www.researchgate.net/profile/Kamal-Acharya-15/publications
Visit google scholar profile : https://scholar.google.com/citations?user=9QoTDzIAAAAJ&hl=en

View publication stats